

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

**THIS PAGE BLANK (USPTO)**

EPODOC / EPO

COPY OF PAPERS  
ORIGINAL

PN - JP10232775 A 19980902  
PD - 1998-09-02  
PR - JP19970052430 19970220  
OPD - 1997-02-20  
TI - PREFETCH MECHANISM  
IN - ITO MASANAO; TAMAOKI YOSHIKO; SUKEGAWA NAONOBU  
PA - HITACHI LTD  
IC - G06F9/38 ; G06F9/38  
- WPI / DERWENT

TI - Pre-fetch mechanism for information processor using semiconductor IC - terminates pre-fetch issue and operation of pre-fetch controller, when difference value output from pre-fetch halt circuit exceeds preset value  
PR - JP19970052430 19970220  
PN - JP10232775 A 19980902 DW199845 G06F9/38 015pp  
PA - (HITA ) HITACHI LTD  
IC - G06F9/38  
AB - J10232775 The pre-fetch mechanism includes an address register (20) in which the data address and address increment of each array component repeatedly accessed by the loop, is stored. The pre- fetch address is generated asynchronous with the execution of a data access instruction to main memory or main cache by a microprocessor, based on the address information stored in the register.

- A request generator (11) issues a pre-fetch request. A pre-fetch halt circuit (13) obtains the difference of the loop count value for loop control accompanied by frequency of pre-fetch instruction execution and the microprocessor. The obtained difference value is compared with a preset value. The pre-fetch issue is terminated and operation of a pre-fetch controller (12) is stopped when the obtained difference value exceeds preset value.

- ADVANTAGE - Improves data access efficiency and calculation capability of microprocessor. Enables optimum pre-fetch for different loop patterns such as double, continuous single fold loops.

- (Dwg.1/14)

OPD - 1997-02-20  
AN - 1998-526737 [45]

- PAJ / JPO

PN - JP10232775 A 19980902  
PD - 1998-09-02  
AP - JP19970052430 19970220  
IN - ITO MASANAO;SUKEGAWA NAONOBU;TAMAOKI YOSHIKO  
PA - HITACHI LTD  
TI - PREFETCH MECHANISM  
AB - PROBLEM TO BE SOLVED: To provide a prefetch mechanism with

which arithmetic ability applied to a microprocessor can be provided at a maximum.

- SOLUTION: Concerning the microprocessor which has a control instruction for loop execution and can execute the instruction of data access to a main memory or a cache, this microprocessor is provided with a prefetch request generating circuit 11, prefetch control circuit 12 and temporary prefetch stop circuit 13. The circuit 11 is provided with a prefetch address array 20 for storing the data addresses and address increments of respective arrangement elements to be repeatedly accessed in a loop, and the circuit 12 prepares a signal L20 successively instructing address read to the array 20 and outputs a signal L21 showing one cycle of read of respective arrangement elements. The circuit 11 issues

**THIS PAGE BLANK (USPTO)**

a prefetch by reading the address corresponding to the signal L20. The circuit 13 subtracts the loop count values of signals L21 and 2 and when the result exceeds a set value, the operation of prefetch control circuit 12 is stopped to temporarily stop the issue of prefetch.

I

- G06F9/38 ;G06F9/38

**THIS PAGE BLANK (USPTO)**

(11)特許出願公開番号

特開平10-232775

(43)公開日 平成10年(1998)9月2日

(51) Int.Cl.<sup>6</sup>  
G 0 6 F 9/38

識別記号  
3 1 0  
3 3 0

F I  
G O 6 F 9/38

3 1 0 A  
3 3 0 E

審査請求 未請求 請求項の数5 FD (全 15 頁)

(21)出願番号 特願平9-52430

(22)出題日 平成9年(1997)2月20日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 伊藤 昌尚

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)発明者 助川 直伸

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

(72)發明者 玉置 由子

東京都国分寺市東恋ヶ窪一丁目280番地

株式会社日立製作所中央研究所内

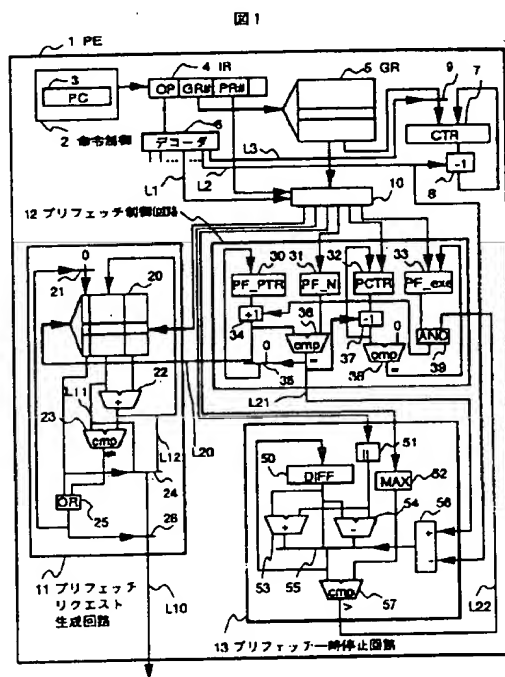
(74) 代理人 弁理士 笹岡 茂 (外1名)

(54) 【発明の名称】 プリフェッチ機構

(57) 【要約】

【課題】 ムドが持つ演算能力を最大限に発揮させるブリフエツチ機構の提供。

【解決手段】 ループ実行のための制御命令を有し、主記憶またはキャッシュへのデータアクセス命令を実行可能なMPにおいて、プリフェッチリクエスト生成回路11と、プリフェッチ制御回路12と、プリフェッチ一時停止回路13を設ける。回路11にはループで繰り返しアクセスされる各配列要素のデータアドレス及びアドレス増分を格納するプリフェッチアドレスアレイ20が設けられ、回路12はアレイ20に対するアドレス読み出しを順次指示する信号L20を作成し、また各配列要素の読み出しが一巡したことを示す信号L21を出力する。回路11は信号L20に応じてアドレスを読み出しプリフェッチを発行する。回路13は信号L21と信号2のループカウント値を引き算し、結果が設定値を越えた場合に前記プリフェッチ制御回路12の動作を中止し、プリフェッチ発行を一時停止する。



## 【特許請求の範囲】

【請求項1】 ループ実行のための制御命令を有し、主記憶またはキャッシュへのデータアクセス命令を実行可能なマイクロプロセッサにおいて、該データアクセス命令を効率的に行うためのプリフェッチ発行のために、プリフェッチ用のデータアドレス、および、アドレス増分を、それぞれ複数組設定可能なレジスタアレイを備え、該レジスタアレイに設定されたアドレス情報に基づいて、該マイクロプロセッサの命令実行とは非同期にプリフェッチのためのアドレスを生成し、前記主メモリにプリフェッチ発行を行う手段と、プリフェッチ発行回数と前記マイクロプロセッサの命令実行に伴うループ制御のためのループカウンタ値を引き算し、該引き算結果を予め設定した値と比較し、該引き算結果が予め設定した値を越えた場合に前記プリフェッチ発行を行う手段の動作を中止し、プリフェッチ発行を一時停止する手段とを備えることを特徴とするプリフェッチ機構。

【請求項2】 請求項1記載のプリフェッチ機構において、前記プリフェッチ用のレジスタアレイは、プリフェッチ用のデータアドレス1つにつき、アドレス増分を2種類以上設定可能とし、前記プリフェッチ発行を行う手段は、プリフェッチ発行回数を計数する2種類以上のカウンタレジスタを設け、該カウンタレジスタの計数値に依存して、該2種類以上設定されたアドレス増分を選択してプリフェッチのためのアドレス生成を行うことを特徴とするプリフェッチ機構。

【請求項3】 請求項1記載のプリフェッチ機構において、前記レジスタアレイに設定されたプリフェッチのためのアドレス情報をグループ化して扱うためのプリフェッチ配列アレイを備え、該プリフェッチ配列アレイは該各グループ化されたプリフェッチのためのアドレス情報に対応して、プリフェッチ回数を設定するためのループ長指示フィールドを有し、前記プリフェッチ発行を行う手段は、前記プリフェッチ配列アレイにおいてグループ化された個々のグループの示す前記レジスタアレイのアドレス情報に基づいて、該グループに対応する前記ループ長指示フィールドの示すプリフェッチ回数だけプリフェッチ発行を行い、該プリフェッチのプリフェッチ発行の実行の後、前記プリフェッチ配列アレイに設定された次のグループに関して同様のプリフェッチ発行を行うことを特徴とするプリフェッチ機構。

【請求項4】 請求項1乃至請求項3のいずれかの請求項記載のプリフェッチ機構であって、プリフェッチ発行に際し主記憶から読み出したデータ

を、プリフェッチ専用データを格納するためのプリフェッチバッファへ格納することを特徴とするプリフェッチ機構。

【請求項5】 請求項1乃至請求項4のいずれかの請求項記載のプリフェッチ機構であって、

プリフェッチ発行に際し前記マイクロプロセッサは、プリフェッチを行うループに関し、ループ制御のための命令またはループカウンタを行うレジスタのいずれかを、プリフェッチ専用として設けることを特徴とするプリフェッチ機構。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、半導体集積回路を用いて構成する情報処理装置に係り、特に、情報処理装置において主記憶アクセスを効率的に行うための機構であるプリフェッチ機構に関する。

## 【0002】

【従来の技術】近年、半導体加工技術の進歩などによって、マイクロプロセッサの動作周波数は向上を続けており、最高速のマイクロプロセッサでは400MHz以上にも達している。これに対して、主記憶を構成する半導体に関しては、その動作周波数が向上はしているものの、マイクロプロセッサの動作周波数の向上よりもゆるやかであるため、主記憶のマイクロプロセッサに対する相対的な動作周波数はむしろ低下する傾向にある。これを補うために、これまではキャッシュと呼ばれる、小容量ではあるが高速の記憶素子を付加して主記憶による性能低下を抑える工夫が行われている。

【0003】しかし、キャッシュは主記憶よりも容量が小さいため、キャッシュの容量を越えるようなデータサイズのプログラムにおいては、キャッシュを単純に付加しただけでは主記憶の動作周波数によって、マイクロプロセッサの処理能力が制限されるという問題点が発生することが知られている。これを回避するための技術として、プリフェッチと呼ばれる方法がある。一般に科学技術計算においては、演算に用いるデータのアドレスを演算開始よりもかなり早期に知ることができるため、演算を行う以前に主記憶からキャッシュにデータを移動することができる。これによって主記憶による制限を回避する技術がプリフェッチである。前もって主記憶からキャッシュにデータを移動する動作を、一般にプリフェッチと呼ぶ。

【0004】近年のマイクロプロセッサは、プリフェッチのための機能を命令として持つことが多い。例えば、IBM社とMotorola社が共同開発したマイクロプロセッサであるPowerPC601にはプリフェッチを行う命令として、dcbt命令等、数種の命令が存在する。これらの命令は、指定されたアドレスの内容を主記憶からキャッシュに格納する働きを持つ(Motorola社発行の「PowerPC601 RISC Microprocessor User's Manual」参照)。このよう



に、プリフェッチ動作を直接に命令で実行する方式を、ソフトウェアプリフェッチ方式と呼ぶ。

【0005】プリフェッチは科学技術計算で多数出現するDOループ内部の配列のアクセスに特に効果を発揮する。DOループ内部の配列のアドレスは規則的に変化することが多いため、配列の先頭アドレス、アドレス増分値などのアドレス情報を用いて、DOループ内部の配列のアドレスをハードウェアが生成することで、自動的に主記憶からキャッシュにデータを移動するプリフェッチ方式も数多く提案されている。このように、プリフェッチの動作を演算などの命令動作とは別個にハードウェアが行う方式をハードウェアプリフェッチ方式と呼ぶ。ハードウェアプリフェッチ方式において、プリフェッチの動作を行うためのアドレス情報を設定する方法としては、ソフトウェアが指示する方式と、ハードウェアが実行中の命令から自動的に検知する方式の2種類がある。前者の例として、Gschwind, M.K. and Pietsch, T.J.: "Vector Prefetching", Computer Architecture News, Special Interest Group on Computer Architecture, Vol. 23, No. 5, December 1995 ACM, pp. 1-7 が挙げられる。

#### 【0006】

【発明が解決しようとする課題】プリフェッチ方式において、プロセッサが持つ演算能力を最大限に発揮させるためには、プロセッサが演算命令を停止することなく動作させるだけの演算データを供給しなければならない。このためには、ループ中の全ての配列について、その配列データが演算に使用されるタイミングよりも一定時間（主記憶のデータを読み出して、キャッシュに格納するために要する時間）だけ前に、プリフェッチ動作を開始する必要がある。これに対して、従来のハードウェアプリフェッチ方式においては、キャッシュのミス率を低減することを主眼としており、必ずしもプロセッサが持つ演算能力を最大限に発揮させることができない。ソフトウェアプリフェッチ方式の場合には、前記dcbt命令などのプリフェッチ命令をループ中に余分に実行する必要があるために、若干のオーバーヘッドを余儀なくされる。また、前記dcbt命令などのプリフェッチ命令を適切にループ中に配置することはコンパイラが行うが、これがコンパイラの他の最適化と整合的に行えるかは一般的には必ずしも両立しないため、やはり、プロセッサが持つ演算能力を最大限に発揮させることは困難である。本発明の目的は、科学技術計算で多数出現するDOループ内部の配列のアクセスに最適なハードウェアプリフェッチ方式を提供し、プロセッサが持つ演算能力を最大限に発揮させることにある。

#### 【0007】

【課題を解決するための手段】上記目的を達成するため、本発明は、ループ実行のための制御命令を有し、主記憶またはキャッシュへのデータアクセス命令を実行可能なマイクロプロセッサにおいて、該データアクセス命

令を効率的に行うためのプリフェッチ発行のために、プリフェッチ用のデータアドレス、および、アドレス増分を、それぞれ複数組設定可能なレジスタアレイを備え、該レジスタアレイに設定されたアドレス情報に基づいて、該マイクロプロセッサの命令実行とは非同期にプリフェッチのためのアドレスを生成し、前記主メモリにプリフェッチ発行を行う手段と、プリフェッチ発行回数と前記マイクロプロセッサの命令実行に伴うループ制御のためのループカウンタ値を引き算し、該引き算結果を予め設定した値と比較し、該引き算結果が予め設定した値を越えた場合に前記プリフェッチ発行を行う手段の動作を中止し、プリフェッチ発行を一時停止する手段とを備えるようにしている。

【0008】また、前記プリフェッチ用のレジスタアレイは、プリフェッチ用のデータアドレス1つにつき、アドレス増分を2種類以上設定可能とし、前記プリフェッチ発行を行う手段は、プリフェッチ発行回数を計数する2種類以上のカウンタレジスタを設け、該カウンタレジスタの計数値に依存して、該2種類以上設定されたアドレス増分を選択してプリフェッチのためのアドレス生成を行うようにしている。

【0009】また、前記レジスタアレイに設定されたプリフェッチのためのアドレス情報をグループ化して扱うためのプリフェッチ配列アレイを備え、該プリフェッチ配列アレイは該各グループ化されたプリフェッチのためのアドレス情報に対応して、プリフェッチ回数を設定するためのループ長指示フィールドを有し、前記プリフェッチ発行を行う手段は、前記プリフェッチ配列アレイにおいてグループ化された個々のグループの示す前記レジスタアレイのアドレス情報に基づいて、該グループに対応する前記ループ長指示フィールドの示すプリフェッチ回数だけプリフェッチ発行を行い、該プリフェッチのプリフェッチ発行の実行の後、前記プリフェッチ配列アレイに設定された次のグループに関して同様のプリフェッチ発行を行うようにしている。

【0010】また、プリフェッチ発行に際し主記憶から読み出したデータを、プリフェッチ専用データを格納するためのプリフェッチバッファへ格納するようにしている。

【0011】プリフェッチ発行に際し前記マイクロプロセッサは、プリフェッチを行うループに関し、ループ制御のための命令またはループカウンタを行うレジスタのいずれかを、プリフェッチ専用として設けるようにしている。

#### 【0012】

##### 【発明の実施の形態】

＜実施例1＞以下、本発明の一実施例を詳細に説明する。図1は、本発明の一実施例であるプリフェッチ機構の構成図である。図1において、1はプロセッサ（PE）であり、2は命令制御、3はプログラムカウンタ

(PC)、4は命令レジスタ(IR)、5は汎用レジスタ(GR)、6は命令デコーダ、7はループカウントレジスタ、8は減算器、9はデータ書込制御回路、10はプリフェッチ情報設定回路である。11はプリフェッチリクエスト生成回路、12はプリフェッチ制御回路、13はプリフェッチ一時停止回路である。20はプリフェッチアドレスアレイ(レジスタアレイで構成してもよい)、21は書込制御回路、22は加算器、23は比較器、24はセクタ、25は論理和回路、26は書込制御回路である。30はプリフェッチポインタレジスタ、31はプリフェッチ配列数レジスタ、32はプリフェッチカウントレジスタ、33はプリフェッチ起動レジスタ、34は加算器、35はセクタ、36は比較器、37は減算器、38は比較器、39は論理積回路である。50はプリフェッチ先行サイクル数カウントレジスタ、51はループサイクル数レジスタ、52はプリフェッチ先行サイクル数指定レジスタ、53は加算器、54は減算器、55はセクタ、56はプリフェッチ先行サイクル数更新回路、57は比較器、L1からL21は信号線である。

【0013】図1ではプロセッサ1内部に命令レジスタ(IR)4、命令デコーダ6がプロセッサ1につき各々1台のみ図示してあるが、複数でもかまわない。すなわち、1サイクルに複数命令を実行可能なスーパースカラ方式のプロセッサでもかまわない。

【0014】次に、このように構成された本システムの動作について、図1を用いて説明する。該プロセッサ(PR)1において、命令制御2内のプログラムカウンタ(PC)3の指定するアドレスの命令を、命令読み出し回路(図示せず)によって命令レジスタ(IR)4に格納する。命令レジスタ(IR)4の命令コードフィールドの内容は、それぞれデコーダ6によって解読され各種命令実行制御回路(図示せず)に指令が行われる。また命令レジスタ(IR)4のレジスタフィールドの内容が指し示す汎用レジスタ(GR)5の内容を読み出して、必要な各種命令実行制御回路(図示せず)に伝達が行われる。このような動作は、従来のマイクロプロセッサの動作と全く同様である。

【0015】以下ではプロセッサ1が主記憶(図示せず)に対してプリフェッチ動作を行う場合について、その処理の経過をプリフェッチリクエスト生成回路11、プリフェッチ制御回路12、プリフェッチ一時停止回路13について説明する。プリフェッチ動作の初期状態の設定は、プロセッサ1が命令によりプリフェッチ情報を格納する資源である、プリフェッチアドレスアレイ20、プリフェッチ配列数レジスタ31(後述の図2、図3の実例では“3”がセットされる)、プリフェッチカウントレジスタ32(上記の実例では“100”がセットされる)、プリフェッチ起動レジスタ33(プリフェッチ起動時に“1”がセットされる)、ループサイクル

数レジスタ51(上記の実例では“3”がセットされる)、プリフェッチ先行サイクル数指定レジスタ52に値を設定(上記の実例では“6”がセットされる)することで行う。すなわち、プリフェッチの初期状態の設定を行う命令が命令レジスタ(IR)4に読み込まれると、デコーダ6によって解読されプリフェッチ情報設定回路10に指示を行い、汎用レジスタ(GR)5の内容を読み出して、命令レジスタ(IR)4中のPR#フィールドで特定されるプリフェッチ情報を格納する資源に対して、値の格納が行われる。また、プリフェッチ動作に関連する資源であるループカウントレジスタ7に対しても、プロセッサ1が命令により値(上記の実例では“100”)を設定する。すなわち、ループカウントレジスタ7の値の設定を行う命令が命令レジスタ(IR)4に読み込まれると、デコーダ6によって解読されデータ書込制御回路9に指示を行い、汎用レジスタ(GR)5の内容を読み出して、ループカウントレジスタ7に対して、値の格納が行われる。なお、プロセッサ1の初期状態においては、プリフェッチポインタレジスタ30、プリフェッチ先行サイクル数カウントレジスタ50には値0が格納される。それ以外の部分の初期状態における値は、従来のマイクロプロセッサと同様な値が格納される。図1においてはプリフェッチ制御のみを示し、それ以外のプロセッサ制御は図示していないが、従来のマイクロプロセッサと同様に処理が行われる。

【0016】プリフェッチのために必要なプリフェッチの初期状態の設定を行う命令が全て実行されると、プリフェッチ動作が開始される。プリフェッチ動作はプリフェッチのためのハードウェア機構であるプリフェッチリクエスト生成回路11、プリフェッチ制御回路12、プリフェッチ一時停止回路13が行い、ループ制御を除くプロセッサ命令の実行とは非同期に行われる。

【0017】プリフェッチのために必要なプリフェッチの初期情報を以下に解説する。第1に、プリフェッチの対象となる配列全てについて、その先頭アドレスとアドレス増分をプリフェッチアドレスアレイ20に設定する。先頭アドレスとはループの最初の繰り返しでアクセスされる各々の配列要素(後述の図2の実例の場合、A(1)、B(1)、C(1))のアドレスであり、アドレス増分とは各々の配列についてのループ繰り返し1回分における配列要素のアドレス変化の大きさ(上記実例では“8”)である。

【0018】プリフェッチアドレスアレイ20の内部構成は図2に示されている。プリフェッチアドレスアレイ20は複数のエントリから成り、各々のエントリは、初回用フィールド、先頭アドレスフィールド、アドレス増分フィールドの3つのフィールドから成る。配列の先頭アドレスは先頭アドレスフィールドに格納され、アドレス増分はアドレス増分フィールドに格納される。初回用フィールドには、先頭アドレスとアドレス増分が格納さ

れる際に、値「1」が格納される。第2に、プリフェッチアドレスアレイ20に設定された配列の種類の数（この場合「3」）。第3に、プリフェッチの対象となるループの繰り返しの回数をプリフェッチカウントレジスタ32に格納する。この値は、プロセッサ1がループ実行を行う際に使用するループカウントレジスタ7に設定される値と同一である（図3の場合「100」）。第4にプリフェッチの対象となるループ1回分の静的な実行サイクル数をループサイクル数レジスタ51に格納する（図3の場合「3」）。静的な実行サイクル数とは、ループ1回分に含まれる命令数を、プロセッサ1が1サイクル当たりに実行可能な命令数で割った値であり、すなわち、主記憶アクセスや分岐命令の実行など、動的な要因が理想的であった場合のループ1回分に要する実行サイクル数である。第5に、プロセッサ1におけるデータ使用に、最大何サイクル先行してプリフェッチを発行するか、そのサイクル数をプリフェッチ先行サイクル数指定レジスタ52に格納する（図3の場合「6」）。最後に、プリフェッチ起動レジスタ33に値「1」を格納すると、以上で設定したプリフェッチ情報に基づいてプリフェッチが起動される。

【0019】これらのプリフェッチ情報の詳細については、下記のアプリフェッチ動作の解説において説明される。プリフェッチ動作は3つの回路、プリフェッチリクエスト生成回路11、プリフェッチ制御回路12、プリフェッチ一時停止回路13によって行われる。プリフェッチリクエスト生成回路11は、プリフェッチ対象となる配列のアドレス情報を持ち、プリフェッチのためのアドレスを生成し、プリフェッチリクエストの発行を行う。プリフェッチ制御回路12は、プリフェッチの回数、プリフェッチの発行タイミングなどを制御する。プリフェッチ一時停止回路13は、プロセッサ1におけるプログラム実行とプリフェッチ発行を監視し、プリフェッチの発行し過ぎを抑制する。

【0020】以下、3つの回路、プリフェッチリクエスト生成回路11、プリフェッチ制御回路12、プリフェッチ一時停止回路13について詳細な説明を行う。プリフェッチリクエスト生成回路11において、プリフェッチの対象となる配列の情報は前述のごとく、プリフェッチに先立ってプリフェッチアドレスアレイ20に格納される。プリフェッチアドレスアレイ20に格納されているどのエントリについてプリフェッチを発行するかは、信号線L20を介してプリフェッチ制御回路12より指定される。該指定されたエントリにおいて、初回用フィールドの値が「1」であった場合には、以下のようにプリフェッチが発行される。プリフェッチアドレスアレイ20から読み出された初回用フィールドの値は論理和回路25に入力される。論理和回路25は書込制御回路26を制御して、セクタ24の出力であるプリフェッチ

アドレスについて信号線L10を介して主記憶（図示せず）に対してプリフェッチを発行する。この際、セクタ24においては初回用フィールドの値が「1」であるため、信号線L11の信号、すなわち、プリフェッチアドレスアレイ20の先頭アドレスフィールドに格納されるアドレスそのままについてプリフェッチが発行される。また、読み出された配列要素の初回用フィールドはOR回路25の出力が「1」のとき「0」に変換される。初回用フィールドの値が「0」であった場合には、以下のようにプリフェッチが発行される。

【0021】プリフェッチアドレスアレイ20から読み出された先頭アドレスフィールドの値とアドレス増分フィールドの値が加算器22で加えられ、元のアドレスフィールドの値と比較器23において比較される。この際、比較の対象となる2つの値は、アドレスの上位部分のみが比較される。ここでアドレスの上位部分とは、プロセッサ1のキャッシュ（図示せず）の1ラインを指定するためのアドレス範囲を指し示す。比較の結果、値が異なる場合には値「1」が論理和回路25に入力される。論理和回路25は書込制御回路26を制御して、セクタ24の出力であるプリフェッチアドレスについて信号線L10を介して主記憶（図示せず）に対してプリフェッチを発行する。この際、セクタ24においては初回用フィールドの値が「0」であるため、信号線L12の信号、すなわち、先頭アドレスフィールドの値とアドレス増分フィールドの値が加算器22で加えられたアドレスについてプリフェッチが発行される。初回用フィールドの値が「0」であって、かつ、比較器23において行われた比較の結果が等しかった場合にはプリフェッチの発行は行われない。

【0022】なお、初回用フィールドの値およびプリフェッチの発行の有無に関わらず、プリフェッチ制御回路12より指定されたエントリについては、先頭アドレスフィールドの値とアドレス増分フィールドの値の加算値が先頭アドレスフィールドに格納される。これにより、プリフェッチアドレスアレイ20の先頭アドレスフィールドは、プリフェッチ対象となる配列について、実際にプロセッサ1が行う命令と同一のアドレス変化を模擬する。プリフェッチの発行はこの模擬の過程で、配列が新たなキャッシュラインをアクセスするタイミングを検出することで行っている。以上のようにして、プリフェッチリクエスト生成回路11において、プリフェッチリクエストが生成される。

【0023】プリフェッチ制御回路12において、プリフェッチのための情報は前述のごとく、プリフェッチに先立ってプリフェッチ配列数レジスタ31、プリフェッチカウントレジスタ32に格納される。プリフェッチ配列数レジスタ31には、プリフェッチを行う配列の個数が、プリフェッチカウントレジスタ32にはプリフェッチの対象となるループの長さが格納される。また、プリ

フェッチ動作の開始前には、プリフェッチポインタレジスタ30は値「0」である。プリフェッチポインタレジスタ30の初期値「0」の格納は、プロセッサ1の初期化時およびプリフェッチ動作の終了時に行われる。全てのプリフェッチ情報が設定された後、プリフェッチ起動レジスタ33に値「1」を設定することでプリフェッチが起動される。

【0024】プリフェッチが起動されるとプリフェッチ制御回路12は以下のように動作を行う。プリフェッチ起動レジスタ33の値が「1」であり、かつ信号線L22を介してプリフェッチ一時停止回路13から伝達される信号値が「1」であるとき、すなわち、後述のように一時停止をしなくてよい状態を示すとき、論理積回路39の出力は値「1」となる。加算器34は論理積回路39の出力が「1」の場合、プロセッサ1の信号クロック（図示せず）に同期して、プリフェッチポインタレジスタ30の値に「1」を加え、その結果をプリフェッチ配列数レジスタ31の値と比較器36において比較する。比較結果が等しくない場合には、セクタ35は加算器34の加算結果をそのまま出力し、比較結果が等しい場合には、値「0」を出力する。セクタ35の出力は、プリフェッチポインタレジスタ30に格納される。また、加算器34の加算結果は信号線L20を介してプリフェッチリクエスト生成回路11に伝達され、前述のごとく、プリフェッチリクエストの生成を行う。また、論理積回路39の出力はプリフェッチリクエスト生成回路11に送られ、もし、論理積回路39の出力が値「0」であった場合には、プリフェッチリクエスト生成回路11に対して、プリフェッチリクエストの生成を行わないよう制御が行われる（制御のための構成は図示省略）。

【0025】本実施例ではプロセッサ1内部に命令レジスタ（IR）4、命令デコーダ6がプロセッサ1につき各々1台の例で解説を行っているが、これらが複数でもかまわない。複数の場合には、プロセッサ1の信号クロック（図示せず）1サイクルにつき、複数命令を実行可能なスーパースカラ方式のプロセッサとなるが、このような場合、すなわち、1サイクルにつき、複数の主記憶アクセス命令をプロセッサ1が実行する場合には、上記信号線L20を介して行うプリフェッチリクエスト生成回路11へのプリフェッチの指示も同様に複数回行い、プリフェッチリクエスト生成回路11において、1サイクルにつき複数のプリフェッチリクエスト生成を行う。

【0026】さて、上記比較器36での比較結果が等しい場合には、信号線L21を介してプリフェッチ一時停止回路13に対し、信号「1」が伝達され、等しくない場合には信号「0」が伝達される。また、比較器36での比較結果が等しい場合には、減算器37においてプリフェッチカウントレジスタ32の値から「1」を引く演算が行われ、その結果はプリフェッチカウントレジスタ32に格納される。また、減算器37の演算結果は比較

器38において値「0」と比較され、等しい場合、つまりあらかじめ設定したループ長の分だけのプリフェッチ動作が行われた場合には、値「0」がプリフェッチ起動レジスタ33に格納され、プリフェッチ動作が完了する。以上のようにして、プリフェッチ制御回路12において、プリフェッチ制御が行われる。

【0027】プリフェッチ一時停止回路13において、プリフェッチのための情報は前述のごとく、プリフェッチに先立ってループサイクル数レジスタ51、プリフェッチ先行サイクル数指定レジスタ52に格納される。ループサイクル数レジスタ51には、プリフェッチの対象となるループに要する静的な実行サイクル数が、プリフェッチ先行サイクル数指定レジスタ52には、プロセッサ1におけるデータ使用に最大何サイクル先行してプリフェッチを発行するか、そのサイクル数が格納される。また、プリフェッチ動作の開始前には、プリフェッチ先行サイクル数カウントレジスタ50は値「0」である。プリフェッチ先行サイクル数カウントレジスタ50の初期値「0」の格納は、プロセッサ1の初期化時およびプリフェッチ動作の終了時に行われる。

【0028】これらの情報が設定された後、プリフェッチ一時停止回路13は以下のように動作を行う。プリフェッチ一時停止回路13は、信号線L2および信号線L21から情報の伝達を受ける。信号線L2は次のような情報を伝達する。命令レジスタ（IR）4にループカウント命令（プログラムにおいてループ単位の終わりに存在する）が格納されこれがデコーダ6によって解説されると信号線L2を介して減算器8に指令が行われ、ループカウントレジスタ7の値が「1」だけ減じられる。このように、信号線L2は、ループカウント命令の実行が1回行われた情報を示す。なお、減算器8の演算結果が「0」であった場合には、命令制御2に対して命令アドレスの指示が行われ、ループ制御が実行される（図示せず）。以下では、信号線L2の値が「1」であった場合、ループカウント命令の実行が行われたものとする。また、前述のように、信号線L21からはプリフェッチ制御回路12より比較器36における比較結果が等しいか否かの情報を伝達される。比較結果が等しい場合には信号「1」が伝達され、等しくない場合には信号「0」が伝達される。信号線L21が値「1」であることは、プリフェッチポインタレジスタ30の内容がプリフェッチカウントレジスタ32に設定された回数だけ加算器34において加算を行われたことを示す。これは、プリフェッチ対象となる配列について、ループ1回分のアドレス変化を模擬したことに等しい。

【0029】さて、このような情報を信号線L2および信号線L21から伝達されて、プリフェッチ一時停止回路13は以下のように動作を行う。セクタ55に対して、次の3つの値が入力される。それらは、プリフェッチ先行サイクル数カウントレジスタ50の値、加算器5

3によってプリフェッチ先行サイクル数カウントレジスタ50の値とループサイクル数レジスタ51の値を加算した値、減算器54によってプリフェッチ先行サイクル数カウントレジスタ50の値からループサイクル数レジスタ51の値を減算した値である。これらの値に対して、セレクト55はプリフェッチ先行サイクル数更新回路56の制御によって値の選択を行う。プリフェッチ先行サイクル数更新回路56には、前述の信号線L2および信号線L21が入力されており、信号線L2が「0」で信号線L21が「1」の場合には、加算器53の出力を選択し、信号線L2が「1」で信号線L21が「0」の場合には、減算器54の出力を選択し、信号線L2、信号線L21がともに「1」の場合、および、信号線L2、信号線L21がともに「0」の場合には、プリフェッチ先行サイクル数カウントレジスタ50の値を選択する。セレクト55の出力はプリフェッチ先行サイクル数指定レジスタ52と比較器57において比較され、セレクト55の出力のほうが大きい場合には、値「0」が信号線L22を介してプリフェッチ制御回路12に伝達され、そうでない場合には値「1」が伝達される。値「0」は一時停止を意味し、信号線L22が値「0」の間はプリフェッチ制御回路12において論理積回路39の出力が「0」となって、これにより、プリフェッチリクエスト生成回路11におけるプリフェッチリクエスト生成が一時停止する。

【0030】この一時停止はプリフェッチリクエストの過剰な生成を抑止するために行われる。この目的は、何らかの要因でプロセッサ1においてループ実行が滞った場合に、プロセッサ1のループ実行に同期してプリフェッチリクエストの生成の一時停止をさせることにある。このような制御を行わない場合、プリフェッチリクエストの過剰な生成のために、プロセッサ1の性能が下がって低下する可能性がある。例えば、プリフェッチリクエストが過剰に生成されたためにプロセッサ1のキャッシュ（図示せず）がプリフェッチされたデータの書込に占有されてしまい、プロセッサ1自身のデータアクセスが阻害される可能性や、プリフェッチリクエストが過剰なために、プロセッサ1がデータアクセスを行う以前に過剰なプリフェッチデータが必要なプリフェッチデータをリフレッシュしてしまう可能性がある。このような状態に陥るのを防ぐために、プリフェッチ一時停止回路13は信号線L2および信号線L21によって、プロセッサ1が実行するループの進み具合と、プリフェッチ機構におけるループの進み具合を比較して、一時停止制御を行う。

【0031】プリフェッチ先行サイクル数カウントレジスタ50はプリフェッチ機構におけるループの進み具合をプロセッサ1のサイクル数（図3の場合、プロセッサクロック数と同等）で管理しており、この値をあらかじめ設定した基準値であるプリフェッチ先行サイクル数指

定レジスタ52と比較して一時停止制御を行う。プリフェッチ先行サイクル数指定レジスタ52には、プロセッサ1が主記憶（図示せず）に対してプリフェッチリクエストを発行してプリフェッチにより得られる主記憶の内容がキャッシュ（図示せず）に格納されるまでのサイクル数に、余裕分のサイクル数を加えた値を設定することが推奨される。プロセッサ1が主記憶（図示せず）に対してプリフェッチリクエストを発行してプリフェッチにより得られる主記憶の内容がキャッシュ（図示せず）に格納されるまでのサイクル数よりも小さい値を設定した場合には、プリフェッチを充分に発行できないためにプロセッサ1の性能が低下する。大きすぎる値を設定した場合には、前述のとおり、プロセッサ1がキャッシュ（図示せず）へのアクセスを阻害される、あるいは、キャッシュ（図示せず）上の必要なデータのリフレッシュが発生するなどの可能性が生ずる。

【0032】なお、本実施例では主記憶（図示せず）から読み出したデータをキャッシュ（図示せず）に格納すると仮定したが、キャッシュ以外のプリフェッチデータを格納するための、キャッシュに類似したバッファであっても構わない。また、プリフェッチ一時停止回路13が一時停止制御に用いるループカウントレジスタ7を、プロセッサ1がループ制御に用いるレジスタとは別個にプリフェッチ動作専用に設けても問題ない。さらに、信号線L2のための情報を生成する命令は、プロセッサ1がループ制御に用いる命令とは別個にプリフェッチ動作専用に設けても問題ない。

【0033】さて、このようにして構成されたプリフェッチ機構の動作の1例を図3を用いて説明する。ここでは、ループ内でA(N)、B(N)、C(N)の3つの配列をN=1から順にアクセスすると仮定しており、ループ長は100を仮定している。これは、プリフェッチアドレスレイ20を図2に示した状態を設定し、プリフェッチ配列数レジスタ31に値「3」を設定し、プリフェッチカウントレジスタ32に「100」を設定した状態に等価である。図3において縦方向は時間の進みを示し、時刻は左端の「プロセッサクロック」の値で表示される。左から第2列の「プリフェッチ対象配列」は、信号線L20によって指示され、プリフェッチリクエスト生成回路11が生成を行ったプリフェッチリクエストを配列要素名(A(1)、B(1)……)で示している。ここでは、キャッシュのラインサイズは配列の1要素の大きさに等しいと仮定した。第3列の「プロセッサ使用配列」は、プロセッサ1が使用した配列要素を示している。主記憶からキャッシュへのレイテンシは6サイクルであると仮定している。これに呼応して、プリフェッチ先行サイクル数指定レジスタ52の値も「6」を設定している。また、プロセッサ1は1サイクルに1命令を実行すると仮定して、ループサイクル数レジスタ51には、値「3」を設定している。これは、ループ内で使

用する配列が3種類であることに対応している。第4列の「L20」は、信号線L20の値を、第5列の「DIFF」は、プリフェッチ先行サイクル数カウントレジスタ50の値を、第6列の「PCTR」はプリフェッチカウントレジスタ32の値を、第7列の「CTR」はループカウントレジスタ7の値を、それぞれ示している。

【0034】「プロセッサクロック」の値が「1」から「6」の範囲では、1サイクルに1配列要素の順でプリフェッチが行われている。これは、プリフェッチポインタレジスタ30の値が、毎サイクル加算器34でインクリメントされ、信号線L20からプリフェッチリクエスト生成回路11内部のプリフェッチアドレスアレイ20に指示が伝達されることで行われる。この際、項目「DIFF」で示されるとおり、プリフェッチ先行サイクル数カウントレジスタ50の値は、プリフェッチ先行サイクル数指定レジスタ52に設定された値「6」を下回っていることが確認できる。また、 $DIFF = (CTR - PCTR) \times 3$ という関係が成り立っていることもわかる。ここに、値「3」はループサイクル数レジスタ51の設定値に対応する。このように、プリフェッチ一時停止回路13においては、信号線L2および信号線L21から、ループカウントレジスタ7およびプリフェッチカウントレジスタ32の情報を得て、該関係に基づいてプリフェッチ先行サイクル数カウントレジスタ50の値を計数し、一時停止制御を行う。

【0035】「プロセッサクロック」の値が「7」から「12」の範囲では、プリフェッチしたデータがプロセッサ1において使用可能となるため、「プロセッサ使用配列」で1サイクルに1配列要素の割合で、プリフェッチした順に使用している（より正しくは、プロセッサ1が使用する順番にプリフェッチを発行し、プリフェッチがその通り行われた結果、プロセッサ1が予定通り順番に使用を行う）。この範囲では、プロセッサ1における配列の使用と、プリフェッチの発行が釣り合っているため、プリフェッチ先行サイクル数カウントレジスタ50の値は一定値「6」となっている。

【0036】「プロセッサクロック」の値が「13」から「15」の範囲では、仮想的にプロセッサ1の実行がストールし、配列の使用が3サイクルの間行われない状況になったものと仮定している。その結果、「プロセッサクロック」が「16」から「18」の範囲ではプリフェッチ先行サイクル数カウントレジスタ50の値が「9」となり、プリフェッチ先行サイクル数指定レジスタ52に設定された値「6」を越える。この結果、プリフェッチ一時停止回路13は信号線L22にプリフェッチを一時停止する信号を出力し、この「プロセッサクロック」の範囲ではプリフェッチが一時停止される。ただし、該範囲においてプロセッサ1ストールは解除され、配列の使用が再開されたとしているために、「プロセッサクロック」の値が「19」以降の範囲では、プリフェ

ッチ先行サイクル数カウントレジスタ50の値は「6」となり、プリフェッチ先行サイクル数指定レジスタ52に設定された値「6」を越えない。この結果、プリフェッチ一時停止回路13は信号線L22にプリフェッチを一時停止する信号を出力するのを取りやめ、プリフェッチ動作が再開される。以上示したようにプリフェッチ動作は、プログラム1のループ実行に同期して発行制御を行うことができ、これによりプリフェッチリクエストの過剰な発行を抑止し、プログラム1にとって最適なプリフェッチを達成する。

【0037】＜実施例2＞次に、本発明の第2の実施例を説明する。本実施例は第1の実施例の変形であり、より複雑なループに対しても効率よくプリフェッチを行うためにプリフェッチ機構を提供することを目的とする。具体的には科学技術計算に頻発する2重ループに対して効率よくプリフェッチを行わせることを目的とする。

【0038】図4は本発明の第2の実施例であるプリフェッチ機構のシステム構成図である。図4中、図1と同一の参照番号を持つ構成要素は第1の実施例の場合と同じ機能を持つ。ここでは図1と異なる構成要素についての説明を行う。図4中、14はプロセッサ（PE）、70はプリフェッチリクエスト生成回路、71はプリフェッチアドレス増分選択回路である。図ではプロセッサ14内部に命令レジスタ（IR）4、命令デコーダ6がプロセッサ1について各々1台のみ図示してあるが、複数でもかまわない。すなわち、1サイクルに複数命令を実行可能なスーパースカラ方式のプロセッサでもかまわない。

【0039】次に、このように構成された本システムの動作について、図4を用いて説明する。命令の読み出し等の動作は第1の実施例と同様であるので省略し、ここでは第1の実施例との相違点である、2重ループに対するプリフェッチ動作を行う場合について、その処理の経過をプリフェッチリクエスト生成回路70、プリフェッチアドレス増分選択回路71、プリフェッチ制御回路12、プリフェッチ一時停止回路13について説明する。

【0040】図5はプリフェッチリクエスト生成回路70の構成図である。図5中、27はプリフェッチアドレスアレイ、28はセクタであり、それ以外の構成要素は図1と同一の参照番号を持ち、第1の実施例の場合と同じ機能を持つ。図7はプリフェッチアドレスアレイ27の構成を示している。プリフェッチアドレスアレイ27は、第1の実施例のプリフェッチアドレスアレイ20の各エントリにアドレス増分フィールドが1つ付け加えられた点が異なっている。以前から存在するアドレス増分フィールドと区別するため、各々、アドレス増分フィールド1、アドレス増分フィールド2と呼んで区別する。これらは、2次元配列の各々の次元に対するアドレス増分を指定するフィールドである。詳細については以下の解説内で述べる。図6はプリフェッチアドレス増分



選択回路71の構成図である。図6中、80は内側ループ長レジスタ、81は内側ループカウントレジスタ、82は減算器、83は比較器、84はセクタである。

【0041】以下、図5から図7もあわせて説明を行う。本実施例における2重ループのアリフェッチのために必要なアリフェッチの初期情報を以下に解説する。まず、アリフェッチの対象となる配列全てについて、その先頭アドレスとアドレス増分をアリフェッチアドレスアレイ27に設定する。先頭アドレスは2重ループのそれぞれのインデックスの最初の繰返しでアクセスされる配列要素のアドレスであり、これを各々の配列について先頭アドレスフィールドに格納する。アドレス増分は2重ループの内側および外側について2種類を設定する。内側ループのループ繰返し1回分における配列要素のアドレス変化の大きさを、アドレス増分フィールド1に格納する。アドレス増分フィールド2には、外側ループのインデックスが変化したときのアドレス変化の大きさ、つまり、ある外側インデックスにおける内側インデックスの最後の配列アクセスのアドレスと、次の外側インデックスにおける内側インデックスの最初の配列アクセスのアドレスとの差を格納する。例えば、以下のような2重ループであれば、配列要素B(11, 1)と配列要素B(1, 2)のアドレスの差をアドレス増分フィールド2に格納する(もちろん、B(11, 2)とB(1, 3)の差でも構わない)。

【0042】DO N=1, 11

DO K=1, 11

S=S+B(K, N)

CONTINUE

CONTINUE

アリフェッチアドレスアレイ27のアドレス増分フィールド2に格納する値は、本実施例に挙げた以外にも様々なアドレスがあり得る。指定の要点は、外側インデックスの変化に対応したアドレス変化に追従できる情報を与えることであり、そのような情報を有していればアリフェッチリクエスト生成回路70において2重ループに対応したアドレス変化を行わせることが可能であり、したがってそのような情報を有しているのであれば、どのようなアドレスでも構わない。アリフェッチアドレスアレイ27の初回用フィールドについては、実施例1と同様である。

【0043】アリフェッチアドレス増分選択回路71内の内側ループ長レジスタ80には、内側ループ長が格納される。内側ループ長は通例、プロセッサ1が内側ループ実行を行う際に、ループカウントレジスタ7に設定される値と同一である。内側ループカウントレジスタ81にも内側ループ長が、内側ループ長レジスタ80設定時に設定される。

【0044】アリフェッチ制御回路12内のアリフェッチ配列数レジスタ31に関しては実施例1と同様であ

り、アリフェッチアドレスアレイ27に設定された配列の種類数を格納する。アリフェッチカウントレジスタ32に格納する値は2重ループの総実行回数、すなわち、内側ループ長×外側ループ長の値を格納する。アリフェッチ一時停止回路13内のループサイクル数レジスタ51に格納する値も実施例1と同様であり、アリフェッチの対象となるループ1回分の静的な実行サイクル数を格納する。アリフェッチ先行サイクル数指定レジスタ52に関しても実施例1と同様である。そして、アリフェッチ制御回路12内のアリフェッチ起動レジスタ33に値「1」を格納すると、以上で設定したアリフェッチ情報に基づいてアリフェッチが起動される点も実施例1と同様である。

【0045】本実施例においては、アリフェッチリクエストの生成方法のみが第1の実施例と異なり、アリフェッチ制御回路12、アリフェッチ一時停止回路13の構成、動作については第1の実施例と同様であるのでこれらの説明は省略し、以下ではアリフェッチリクエスト生成回路70とアリフェッチアドレス増分選択回路71の動作についてのみ説明を行う。アリフェッチリクエスト生成回路70において、アリフェッチの対象となる配列の情報は前述のごとく、アリフェッチに先立ってアリフェッチアドレスアレイ27に格納される。アリフェッチアドレスアレイ27に格納されているどのエントリについてアリフェッチを発行するかは、信号線L20を介してアリフェッチ制御回路12より指定される。これは実施例1と同様である。該指定されたエントリから読み出されたアドレス増分フィールド1とアドレス増分フィールド2の値はセクタ28に入力され、アリフェッチアドレス増分選択回路71からの信号線L23の値によって選択されて加算器22に入力される。加算器22に入力されて以降は、実施例1と同様の動作である。

【0046】信号線L23はアリフェッチアドレス増分選択回路71から以下のように生成される。アリフェッチアドレス増分選択回路71には、アリフェッチ制御回路12から信号線L21が入力される。信号線L21は実施例1で解説したように、内側ループ1回分のアリフェッチリクエスト生成を信号線L20を介してアリフェッチリクエスト生成回路70に指示する度に値「1」が出力される。信号線L21が値「1」を取ると、減算器82は内側ループカウントレジスタ81に格納された値をデクリメントする。該デクリメントされた値は比較器83に入力され、値「0」と比較される。比較結果が等しい場合には、セクタ84は内側ループ長レジスタ80の値を選択し、等しくない場合には、該デクリメントされた値を選択し、選択された値を内側ループカウントレジスタ81に格納する。該比較結果が等しい場合には、信号線L23を介して、アリフェッチリクエスト生成回路70においてアドレス増分フィールド2の値がセクタ28によって選択され、該比較結果が等しくない

場合には、アドレス増分フィールド1の値がセクタ28によって選択される。このように信号線L23は内側ループ長に相当する回数の内側ループの実行完了を示しており、この情報によってプリフェッチリクエスト生成回路70は、アドレス増分を切り換える。

【0047】以上のようにして、本実施例では2重ループに対応したプリフェッチアドレスを自動的に生成することができ、したがって2重ループの実行においてもプロセッサ1はループ実行の初回のメモリレイテンシ以外のレイテンシを隠蔽することができる。なお、本実施例では主記憶（図示せず）から読み出したデータをキャッシュ（図示せず）に格納すると仮定したが、キャッシュ以外のプリフェッチデータを格納するための、キャッシュに類似したバッファであっても構わない。また、プリフェッチ一時停止回路13が一時停止制御に用いるループカウンタレジスタ7を、プロセッサ1がループ制御に用いるレジスタとは別個にプリフェッチ動作専用に設けても問題ない。さらに、信号線L2のための情報を生成する命令は、プロセッサ1がループ制御に用いる命令とは別個にプリフェッチ動作専用に設けても問題ない。

【0048】＜実施例3＞次に、本発明の第3の実施例を説明する。本実施例は第1の実施例の変形であり、第2の実施例とは別の形の複雑なループに対しても効率よくプリフェッチを行うためにプリフェッチ機構を提供することを目的とする。具体的には異なる複数の1重ループが連続して実行される場合に対して効率よくプリフェッチを行わせることを目的とする。

【0049】図8は本発明の第3の実施例であるプリフェッチ機構のシステム構成図である。図8中、図1と同一の参照番号を持つ構成要素は第1の実施例の場合と同じ機能を持つ。ここでは図1と異なる構成要素についてのみ説明を行う。図8中、15はプロセッサ（PE）、72はプリフェッチ制御回路、73はプリフェッチ上位制御回路、L25からL29は信号線である。図ではプロセッサ15内部に命令レジスタ（IR）4、命令デコーダ6がプロセッサ1各々1台のみ図示してあるが、複数でもかまわない。すなわち、1サイクルに複数命令を実行可能なスーパースカラ方式のプロセッサでもかまわない。

【0050】次に、このように構成された本システムの動作について、図8を用いて説明する。命令の読み出し等の動作は第1の実施例と同様であるので省略し、ここでは第1の実施例との相違点である、連続する1重ループに対するプリフェッチ動作を行う場合について、その処理の経過をプリフェッチリクエスト生成回路11、プリフェッチ制御回路72、プリフェッチ上位制御回路73、プリフェッチ一時停止回路74について説明する。

【0051】図9はプリフェッチ制御回路72の構成図である。図9中、90はプリフェッチアドレスアレイ先頭エントリ指定レジスタ、91は加算器であり、それ以

外の構成要素は図1と同一の参照番号を持ち、第1の実施例の場合と同じ機能を持つ。図10はプリフェッチ上位制御回路73の構成図である。図10中、100はプリフェッチ配列アレイ、101はプリフェッチ起動レジスタ、102は減算器、103は論理和回路、L40は信号線である。図11はプリフェッチ配列アレイ100の構成を示している。プリフェッチ配列アレイ100は複数のエントリから成り、各々のエントリは、有効フィールド、先頭エントリ指示フィールド、末尾エントリ指示フィールド、ループ長指示フィールド、ループサイクル数指示フィールドの5つのフィールドから成る。図14はプリフェッチ一時停止回路74の構成図である。図14中、L30は信号線であり、それ以外の構成要素は図1と同一の参照番号を持ち、第1の実施例の場合と同じ機能を持つ。

【0052】本実施例における、連続する1重ループのプリフェッチのために必要なプリフェッチの初期情報を以下に解説する。まず、プリフェッチの対象となるループに関して、それらループ内の配列について、その先頭アドレスとアドレス増分をプリフェッチアドレスアレイ20に設定する。設定は、連続する複数のループについてまとめて行ってよい。このとき、同じループ内の配列については、プリフェッチアドレスアレイ20の連続したエントリに設定を行う。さらに、各々のループについて、ループ内の配列のアドレス情報をセットしたプリフェッチアドレスアレイ20のエントリの先頭と末尾を、プリフェッチ上位制御回路73内のプリフェッチ配列アレイ100の先頭エントリ指示フィールドと末尾エントリ指示フィールドに設定する。また、そのループのループ長をループ長指示フィールドに、ループ1回分の静的な実行サイクル数をループサイクル数指示フィールドに設定する。プリフェッチ配列アレイにおいて配列情報が設定されたエントリの有効フィールドには値「1」が設定される。なお、プロセッサ1の初期状態においては、全ての有効フィールドには値「0」が設定される。残るループに関しては、プロセッサ1がループ実行を行って、設定されたプリフェッチを終了した後、プリフェッチアドレスアレイ20の開放されたエントリ、および、プリフェッチ配列アレイ100に対して上記と同様行う。詳しくは、後述する。なお、プリフェッチアドレスアレイ20の初回用フィールド、プリフェッチポインタレジスタ30、プリフェッチ先行サイクル数指定レジスタ52については、実施例1と同様である。

【0053】上記初期設定が終了した後、プリフェッチ起動レジスタ101に値「1」を格納すると、以上で設定したプリフェッチ情報に基づいてプリフェッチが起動される。起動されると、まず、プリフェッチ配列アレイ100において最も過去に設定された有効なエントリの内容が読み出され、先頭エントリ指示フィールドの値は-1され（図示せず）、この-1された値は信号線L2



5を介してプリフェッチ制御回路72内のプリフェッチアドレスレイ先頭エントリ指定レジスタレジスタ90に格納される。末尾エントリ指示フィールドの値は減算器102によって先頭エントリ指示フィールドの値から-1した値が減ぜられた後、信号線L26を介してプリフェッチ制御回路72内の、プリフェッチ配列数レジスタ31に格納される。ループ長指示フィールドの値は信号線L27を介してプリフェッチ制御回路72内のプリフェッチカウンタレジスタ32に格納される。ループサイクル数指示フィールドの値は信号線L30を介してプリフェッチ一時停止回路74内のループサイクル数レジスタ51に格納される。

【0054】プリフェッチリクエスト生成回路11の動作は実施例1と同様であり、プリフェッチ一時停止回路74の動作も実施例1のプリフェッチ一時停止回路13と同様であるので、以下では、プリフェッチ上位制御回路73、プリフェッチ制御回路72について説明を行う。プリフェッチ制御回路72の動作も大部分が実施例1のプリフェッチ制御回路12と同様であり、以下では異なる部分についてのみ説明を行う。本実施例では、信号線L20の出力が加算器34の出力にプリフェッチアドレスレイ先頭エントリ指定レジスタレジスタ90の値を加算器91で加算している。ただし、加算器91における加算においては、加算の結果がプリフェッチアドレスレイ20のエントリ数の値を超えた場合には、該結果から該プリフェッチアドレスレイ20のエントリ数を減じるものとする。

【0055】また、比較器38の出力結果は信号線L29を介してプリフェッチ上位制御回路73内のプリフェッチ起動レジスタ101に入力される。比較器38における比較が値「0」と等しい場合、つまりあらかじめ設定したループ長の分だけのプリフェッチ動作が行われた場合には、プリフェッチ配列アレィ100の1つのエントリにおけるプリフェッチ動作が完了したことを示す。この場合、フェッチ配列アレィ100において最も古いエントリ、すなわち、プリフェッチ動作の完了したエントリの有効フィールドに値「0」が格納され、次に古い有効なエントリの内容が読み出され、先頭エントリ指示フィールドの値から-1された値は信号線L25を介してプリフェッチ制御回路72内のプリフェッチアドレスレイ先頭エントリ指定レジスタレジスタ90に格納され、末尾エントリ指示フィールドの値は減算器102によって先頭エントリ指示フィールドの値から-1した値が減ぜられた後、信号線L26を介してプリフェッチ制御回路72内の、プリフェッチ配列数レジスタ31に格納され、ループ長指示フィールドの値は信号線L27を介してプリフェッチ制御回路72内のプリフェッチカウンタレジスタ32に格納され、ループサイクル数指示フィールドの値は信号線L30を介してプリフェッチ一時停止回路74内のループサイクル数レジスタ51に格納

される。そして、プリフェッチ動作がこれらのプリフェッチ情報に基づいて継続される。

【0056】さて、このようにして、プリフェッチ動作を行って全てのエントリのプリフェッチが完了した場合、各有効フィールドの値が信号線L40を介して論理和回路103に入力されており、この結果、該論理和回路103の出力が値「0」となり、これがプリフェッチ起動レジスタ101に設定されて、一連のプリフェッチ動作が終了する。

【0057】さて、このようにして構成されたプリフェッチ機構の動作の1例を図12および図13を用いて説明する。図12の左側に動作例のためのプログラムが示されており、右側にはこのプログラムにプリフェッチ情報の設定を行う命令を挿入した結果が示されている。プリフェッチ情報の設定は、「prefetch」というオペコードにプリフェッチ対象となる配列名をオペランドとして、発行タイミングのみを示している。図12中、(1)から(6)まで番号を付加した文について、その文を実行後のプリフェッチ配列アレィ100およびプリフェッチアドレスレイ20の状態を図13に示している。ここで、プリフェッチ配列アレィ100のエントリ数は3であり、プリフェッチアドレスレイ20のエントリ数は8であると仮定している。また、プリフェッチ配列アレィ100において、ループサイクル数指示フィールドは省略して図示していない。

【0058】①において、最初のループのために配列A、B、Cのプリフェッチ情報を設定している。すなわち、プリフェッチ配列アレィ100の先頭エントリに、プリフェッチアドレスレイ20のエントリ1番から3番を使用し、ループ長が100であることを登録している。そして、プリフェッチアドレスレイ20のエントリ1番には配列Aの先頭アドレスと、そのアドレス増分「8」を、エントリ2番には配列Bの先頭アドレスと、そのアドレス増分「8」を、エントリ3番には配列Cの先頭アドレスと、そのアドレス増分「8」を登録している。

【0059】引き続いて行われる②において、次のループのための配列D、E、Fのプリフェッチ情報を、最初のループの実行に先立って設定している。プリフェッチ配列アレィ100の2番目のエントリに、プリフェッチアドレスレイ20のエントリ4番から6番を使用し、ループ長が150であることを登録している。そして、プリフェッチアドレスレイ20の対応するエントリに配列D、E、Fのプリフェッチ情報を登録している。

【0060】③では最初のループの実行が終了し、(2)においてはプリフェッチ配列アレィ100の先頭エントリに登録されていた配列A、B、Cのプリフェッチ情報が完了/抹消され、代わって配列D、E、Fのプリフェッチ情報が先頭エントリに移動している。配列

A、B、Cのプリフェッチ情報が完了に引き続いて、配列D、E、Fのプリフェッチ動作が起動される。これにより、③の後に配列D、E、Fのプリフェッチ情報の設定・起動を行うよりも早期に配列D、E、Fのプリフェッチ動作を開始できる。このようにして、ループが連続している場合の性能向上を達成できる。

【0061】④において、配列U、V、Wのプリフェッチ情報の設定が行われる。このとき、プリフェッチ配列アレイ100のエントリ7番、8番、1番に登録を行っている。このように、プリフェッチ配列アレイ100のエントリの8番から1番は連続したエントリとして用いられる。また、エントリ1番は①において配列Aのプリフェッチ情報を格納していたが、③の段階では既に配列Aのプリフェッチ動作を完了しているため、④では再利用することが可能な状態となっている。同様に、⑤においてはプリフェッチアドレスアレイ20のエントリ4番から6番に登録されていた配列D、E、Fのプリフェッチは完了してしており、⑥で配列X、Y、Zを登録する際にエントリ4番の再利用を行っている。

【0062】このように、プリフェッチ配列アレイ100、プリフェッチアドレスアレイ20のエントリの再利用を行うことができるため、プリフェッチアドレスアレイ20のエントリ数を越える配列を使用するループに対しても、本実施例であればプリフェッチ動作を行うことが可能となる。すなわち、該エントリ数を越える配列を使用するループを、小ループに分割してプリフェッチを行えばよい。このとき、連続する2個のループで使用する配列数をプリフェッチアドレスアレイ20のエントリ数以内とすれば、図12、図13で説明した例のようにプリフェッチを使用することが可能となる。

【0063】以上示したように、本実施例によって複数のループ実行に対して複数のプリフェッチ情報の設定を、プリフェッチ配列アレイ100およびプリフェッチアドレスアレイ20のエントリ数の範囲内で前もって行うことが可能になる。これにより、該複数のプリフェッチ情報は各々が終了する毎に連続して動作を行うことが可能になり、複数のループ実行を切れ目無く動作させることが可能になる。また、プリフェッチアドレスアレイ20のエントリ数を越える配列を使用するループに対しても、小ループに分割してプリフェッチを行うことで、効率よくプリフェッチ動作を使用することが可能となる。

【0064】なお、本実施例では主記憶（図示せず）から読み出したデータをキャッシュ（図示せず）に格納すると仮定したが、キャッシュ以外のプリフェッチデータを格納するための、キャッシュに類似したバッファであっても構わない。また、プリフェッチ一時停止回路74が一時的停止制御に用いるループカウンレジスタ7を、プロセッサ1がループ制御に用いるレジスタとは別個に

プリフェッチ動作専用に設けても問題ない。さらに、信号線L2のための情報を生成する命令は、プロセッサ1がループ制御に用いる命令とは別個にプリフェッチ動作専用に設けても問題ない。また、本実施例と実施例2を組み合わせ、連続する2重ループ、または連続する2重ループと1重ループに対してプリフェッチを行う機構を構築することが出来ることは言うまでもない。

【0065】

【発明の効果】本発明によれば、マイクロプロセッサに組み込んだプリフェッチ機構によって、マイクロプロセッサが必要とするデータを主記憶からタイミングよくキャッシュに格納することが可能となり、マイクロプロセッサが演算命令を停止することなく動作することができる。また、マイクロプロセッサの実行に同期して過剰なプリフェッチを抑制することができるため、プリフェッチ動作とマイクロプロセッサの命令実行の競合を抑止し、マイクロプロセッサが持つ演算能力を最大限に発揮させることができる。さらに、2重ループや連続する1重ループなど多様なパターンのループに対して最適なプリフェッチが可能となる。

【図面の簡単な説明】

【図1】本発明の第1の実施例に係るプリフェッチ機構の構成を示す図である。

【図2】本発明の第1の実施例に係るプリフェッチ機構におけるプリフェッチアドレスアレイの構成を示す図である。

【図3】本発明の第1の実施例に係るプリフェッチ機構の動作例を示す図である。

【図4】本発明の第2の実施例に係るプリフェッチ機構の構成を示す図である。

【図5】本発明の第2の実施例に係るプリフェッチ機構におけるプリフェッチリクエスト生成回路の構成を示す図である。

【図6】本発明の第2の実施例に係るプリフェッチ機構におけるプリフェッチアドレス増分選択回路の構成を示す図である。

【図7】本発明の第2の実施例に係るプリフェッチ機構におけるプリフェッチアドレスアレイの構成を示す図である。

【図8】本発明の第3の実施例に係るプリフェッチ機構の構成を示す図である。

【図9】本発明の第3の実施例に係るプリフェッチ機構におけるプリフェッチ制御回路の構成を示す図である。

【図10】本発明の第3の実施例に係るプリフェッチ機構におけるプリフェッチ上位制御回路の構成を示す図である。

【図11】本発明の第3の実施例に係るプリフェッチ機構におけるプリフェッチ配列アレイの構成を示す図である。

【図12】本発明の第3の実施例に係るプリフェッチ機

構の動作例の対象となるサンプルプログラムを示す図である。

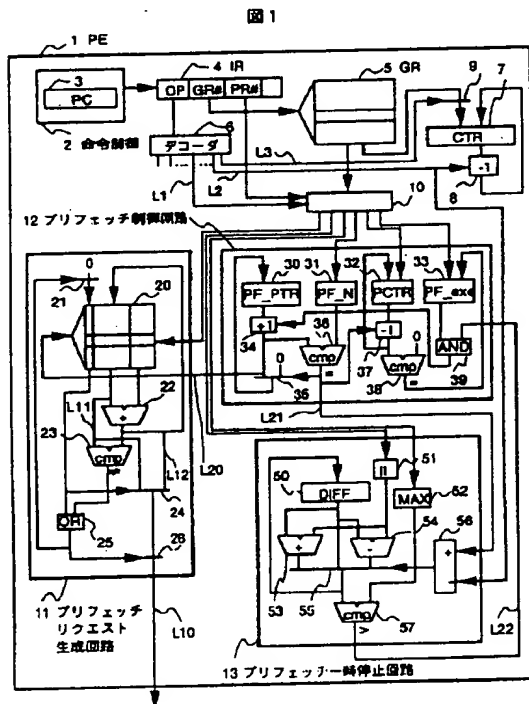
【図13】本発明の第3の実施例に係るプリフェッチ機構の動作例を示す図である。

【図14】本発明の第3の実施例に係るプリフェッチ機構におけるプリフェッチ一時停止回路の構成を示す図である。

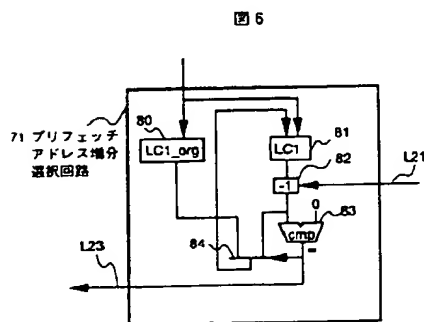
【符号の説明】

- 1 プロセッサ (PE)
- 2 命令制御
- 3 プログラムカウンタ (PC)
- 4 命令レジスタ (IR)
- 5 汎用レジスタ (GR)
- 6 命令でコード
- 7 ループカウンタレジスタ
- 8 減算器
- 9 データ書き込み制御回路
- 10 プリフェッチ情報設定回路
- 11、70 プリフェッチリクエスト生成回路
- 12、72 プリフェッチ制御回路
- 13、74 プリフェッチ一時停止回路
- 71 プリフェッチアドレス増分選択回路
- 73 プリフェッチ上位制御回路

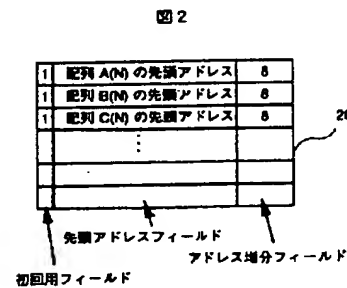
【図1】



【図6】



【図2】



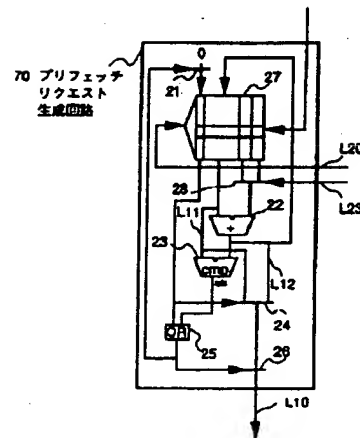
【図3】

図 3

プロセッサ ブロック	プリフェッチ 対象配列	プロセッサ 使用配列	L20	DIFF	PCTR	CTR
1	A(1)		1	0	100	100
2	B(1)		2	0	100	100
3	C(1)		3	0	100	100
4	A(2)		1	3	99	100
5	B(2)		2	3	99	100
6	C(2)		3	3	99	100
7	A(3)	A(1)	1	6	98	100
8	B(3)	B(1)	2	6	98	100
9	C(3)	C(1)	3	6	98	100
10	A(4)	A(2)	1	6	97	99
11	B(4)	B(2)	2	6	97	99
12	C(4)	C(2)	3	6	97	99
13	A(5)		1	6	96	98
14	B(5)	ストール	2	6	96	98
15	C(5)		3	6	96	98
16		A(3)	-	9	95	98
17		B(3)	-	9	95	98
18		C(3)	-	9	95	98
19	A(6)	A(4)	1	6	95	97
20	B(6)	B(4)	2	6	95	97
21	C(6)	C(4)	3	6	95	97
22	A(7)	A(5)	1	6	94	96
23	B(7)	B(5)	2	6	94	96
24	C(7)	C(5)	3	6	94	96

【图5】

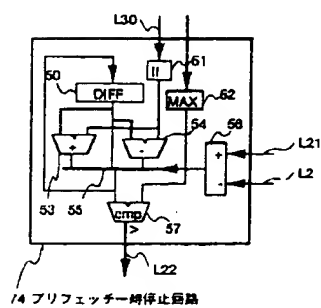
**图 5**



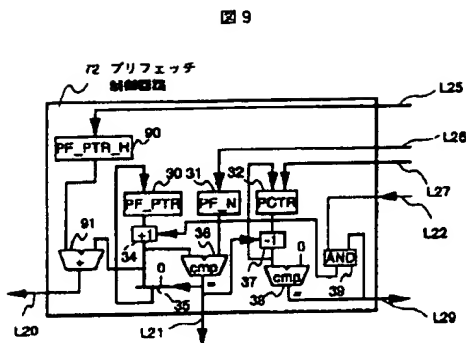
【図8】

图 8

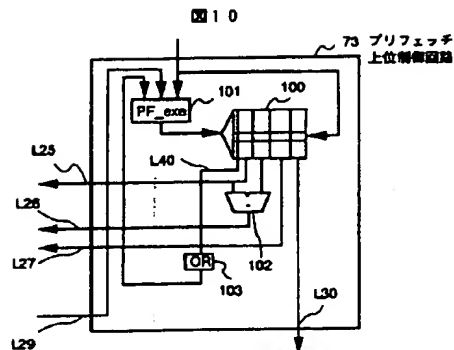
图 14



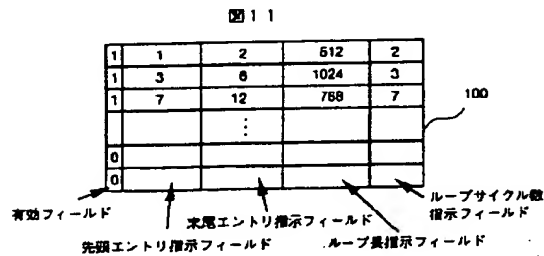
【図9】



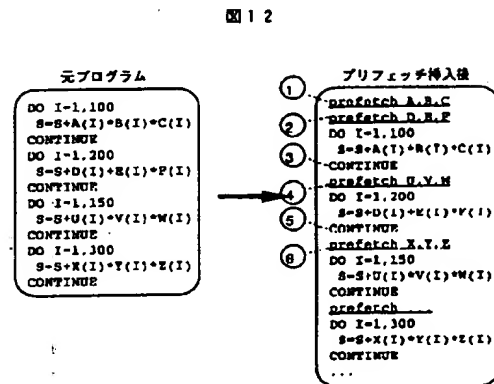
【図10】



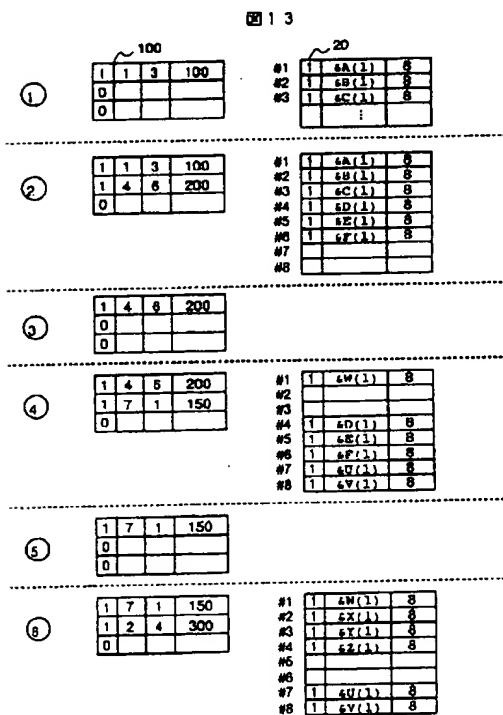
【図11】



【図12】



【図13】



**THIS PAGE BLANK (USPTO)**